

Analiza obiektowa

3@KASK

16 czerwca 2009

Symbol projektu: 3@KASK	Opiekun projektu: mgr inż. Tomasz Boiński
Nazwa Projektu: Wizualizacja grafów za pomocą biblioteki Prefuse	

Nazwa Dokumentu: Analiza obiektowa	Nr wersji: 2.0
Odpowiedzialny za dokument: Piotr Kunowski	Data pierwszego sporządzenia: 23 maja 2009
Przeznaczenie: DLA KLIENTA	Data ostatniej aktualizacji: 16 czerwca 2009

Historia dokumentu

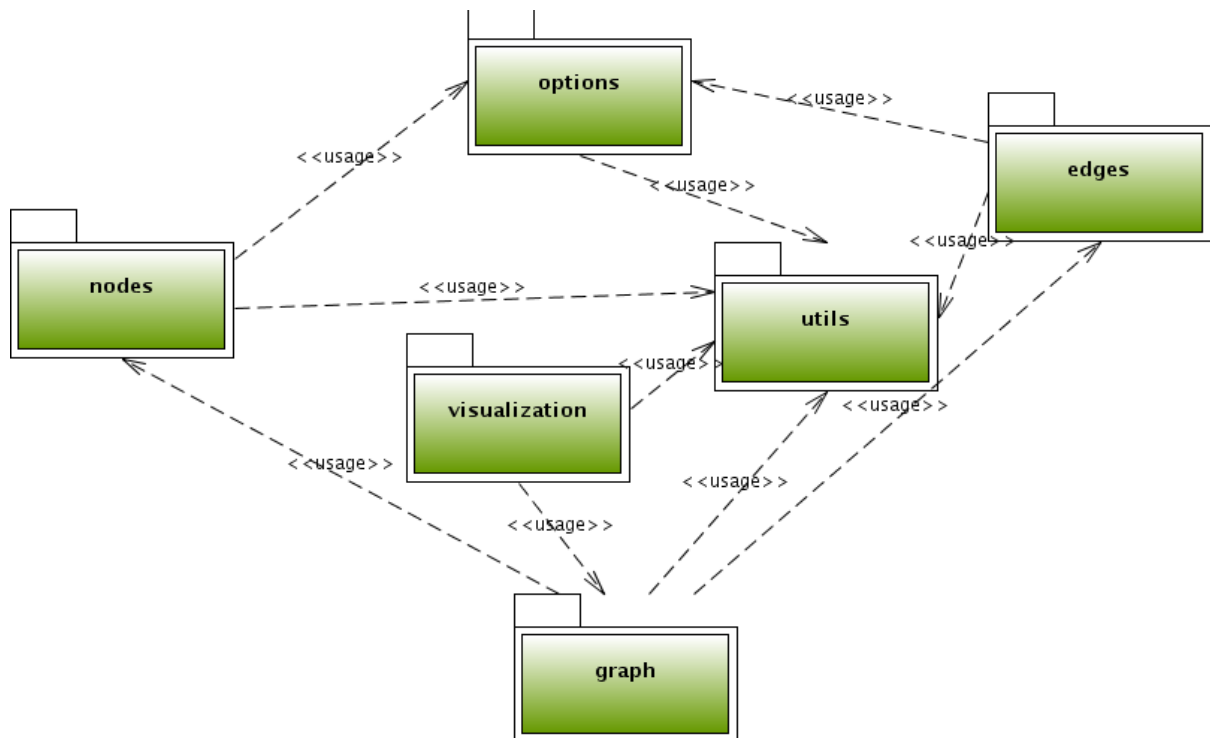
Wersja	Opis modyfikacji	Rozdział/strona	Autor modyfikacji	Data
1	Stworzenie	wszystkie	Grupa projektowa	23.05.09
1.1	Dodano pakiet Utils	1, 3	Anna Jaworska	2.06.09
2	Dodano zaktualizowane diagramy oraz opisy klas	wszystkie	Grupa projektowa	16.06.09

Spis treści

1	Pakiety	3
1.1	Diagram	3
1.2	Opis pakietów	3
2	Pakiet options	4
2.1	Diagram	4
2.2	Opis klasy	4
3	Pakiet nodes	7
3.1	Diagram	7
3.2	Opis klasy	7
4	Pakiet edges	13
4.1	Diagram	13
4.2	Opis klasy	13
5	Pakiet visualization	16
5.1	Diagram	16
5.2	Opis klasy	16
6	Pakiet graph	18
6.1	Diagram	18
6.2	Opis klasy	18
7	Pakiet utils	19
7.1	Diagram	19
7.2	Opis klasy	19

1 Pakiety

1.1 Diagram



1.2 Opis pakietów

P001	options
Opis:	Pakiet zawierający klasy z polami opisującymi różne (modyfikowalne) ustawienia wizualizacji takie jak: kolory, grubość linii itp.
Interfejsy:	
Realizowane wymagania:	WF002, WF001, WI004
Priorytet:	średnio ważne

P002	nodes
Opis:	Pakiet z klasami odpowiedzialnymi za wizualizację i przechowywanie danych o wierzchołkach.
Interfejsy:	
Realizowane wymagania:	WF004, WF005, WF006, WF007, WI004
Priorytet:	bardzo ważne

P003	edges
Opis:	Pakiet z klasami odpowiedzialnymi za wizualizację i przechowywanie danych o krawędziach.
Interfejsy:	
Realizowane wymagania:	WF006, WF007, WI004
Priorytet:	bardzo ważne


P004	visualization
Opis:	Zawiera dodatkowe klasy przydatne w wizualizacji.
Interfejsy:	
Realizowane wymagania:	WF001, WF008, WI004
Priorytet:	średnio ważne


P005	graph
Opis:	Pakiet zawiera klasy, które zawierają podstawowe operacje na danych OwlApi oraz graph.
Interfejsy:	
Realizowane wymagania:	WD001
Priorytet:	bardzo ważne

P006	utils
Opis:	Pakiet zawiera klasy pomocnicze
Interfejsy:	
Realizowane wymagania:	CF005
Priorytet:	bardzo ważne

2 Pakiet options

2.1 Diagram

 NodeColors
<i>Attributes</i>
protected Color allValuesFromNodeColor
protected Color anonymousClassNodeColor
protected Color cardinalityNodeColor
protected Color cardinalityValueNodeColor
protected Color classNodeColor
protected Color complementOfNodeColor
protected Color dataTypeNodeColor
protected Color differentNodeColor
protected Color functionalPropertyNodeColor
protected Color individualNodeColor
protected Color informationNodeColor
protected Color intersectionOfNodeColor
protected Color inverseFunctionalPropertyColor
protected Color maxCardinalityValueNodeColor
protected Color minCardinalityValueNodeColor
protected Color nothingNodeColor
protected Color oneOfNodeColor
protected Color propertyNodeColor
protected Color sameAsNodeColor
protected Color someValuesFromNodeColor
protected Color symmetricPropertyNodeColor
protected Color thingNodeColor
protected Color transitivePropertyNodeColor
protected Color unionOfNodeColor

 EdgeColors
<i>Attributes</i>
protected Color rangeEdgeColor
protected Color domainEdgeColor
protected Color edgeColor
protected Color equivalentEdgeColor
protected Color equivalentPropertyEdgeColor
protected Color functionalEdgeColor
protected Color inverseOfEdgeColor
protected Color propertyEdgeColor
protected Color subEdgeColor

2.2 Opis klasy

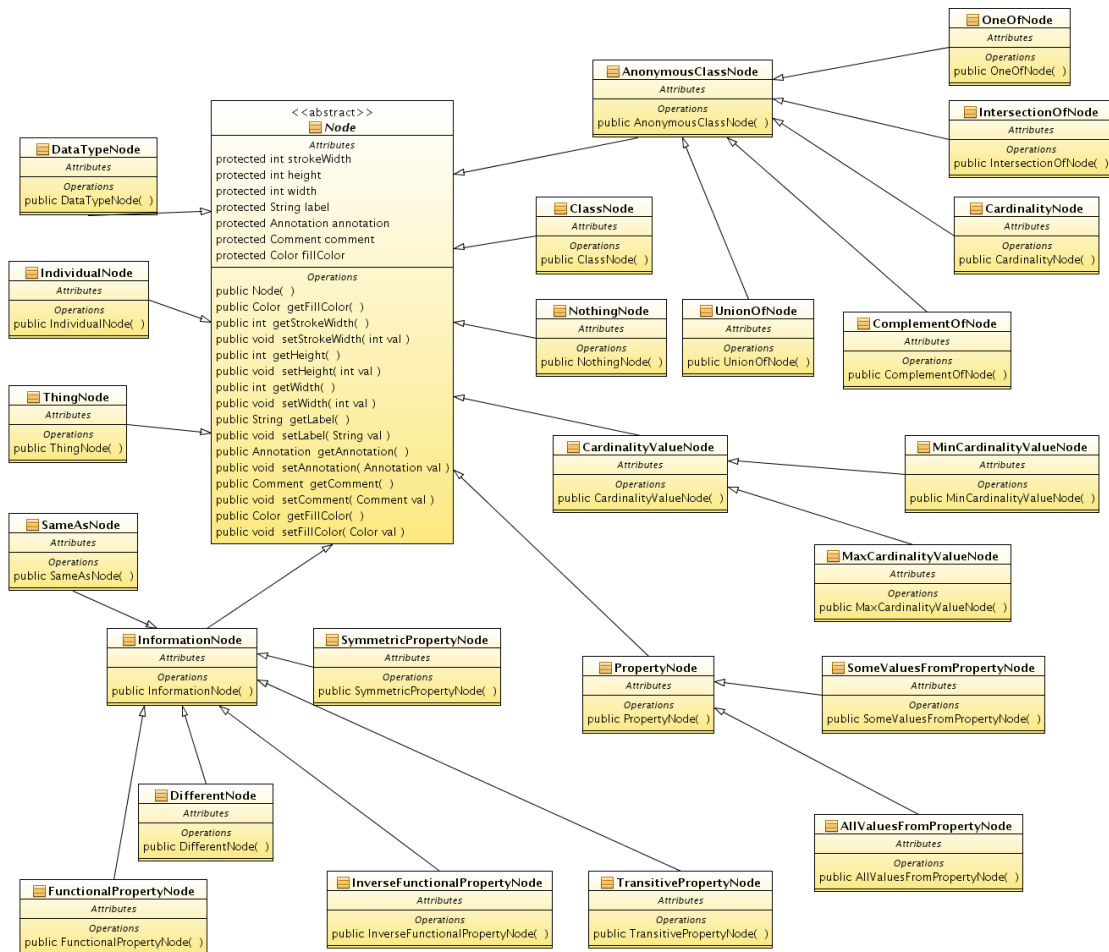
CO001	EdgeColors
Opis:	Zawiera definicje kolorów dla poszczególnych rodzajów krawędzi.
Klasy nadrzędne:	
Atrybuty:	<ul style="list-style-type: none"> • domainEdgeColor • edgeColor • equivalentEdgeColor • equivalentPropertyEdgeColor • functionalEdgeColor • inverseOfEdgeColor • propertyEdgeColor • rangeEdgeColor • subEdgeColor
Metody:	
Realizowane wymagania:	WF002
Priorytet:	średnio ważny

CO002	NodeColors
Opis:	Zawiera definicje kolorów dla poszczególnych rodzajów krawędzi.
Klasy nadrzędne:	

Atrybuty:	<ul style="list-style-type: none"> • allValuesFromNodeColor • cardinalityNodeColor • cardinalityValueNodeColor • classNodeColor • complementOfNodeColor • dataTypeNodeColor • differentNodeColor • functionalPropertyNodeColor • individualNodeColor • informationNodeColor • intersectionOfNodeColor • inverseFunctionalNodeColor • maxCardinalityValueNodeColor • minCardinalityValueNodeColor • nothingNodeColor • oneOfNodeColor • propertyNodeColor • sameAsNodeColor • someValuesFromNodeColor • symmetricPropertNodeColor • thingNodeColor • transitivePropertyNodeColor • unionOfNodeColor
Metody:	
Realizowane wymagania:	WF002
Priorytet:	średnio ważny

3 Pakiet nodes

3.1 Diagram



3.2 Opis klasy

CN001	Node
Opis:	Klasa nadrzędna względem wszystkich klas obsługi wierzchołków. Zawiera definicje podstawowych atrybutów i metod.
Klasy nadrzędne:	
Atrybuty:	<ul style="list-style-type: none"> strokeWidth height width annotation comment Color fillColor String label
Metody:	

Realizowane wymagania:	WF004, WF005, WF006, WF007, WI004
Priorytet:	bardzo ważne

CN002	AllValuesFromPropertyNode
Opis:	Klasa reprezentuje wierzchołek, będący OWL Property typu AllValuesFrom.
Klasy nadrzędne:	Node
Atrybuty:	
Metody:	
Realizowane wymagania:	WF004, WF006, WF007, WI004
Priorytet:	ważne

CN003	AnonymousClassNode
Opis:	Klasa reprezentuje wierzchołek klas anonimowych OWL.
Klasy nadrzędne:	Node
Atrybuty:	
Metody:	
Realizowane wymagania:	WF005, WI004
Priorytet:	ważne

CN004	CardinalityNode
Opis:	Klasa reprezentuje wierzchołek klas anonimowych OWL będących wynikiem ograniczenia kardynalności.
Klasy nadrzędne:	AnonymousNode
Atrybuty:	
Metody:	
Realizowane wymagania:	WF007, WI004
Priorytet:	ważne

CN005	CardinalityValueNode
Opis:	Klasa reprezentuje wierzchołek z dokładnym ograniczeniem kardynalności (OWL Cardinality).
Klasy nadrzędne:	Node
Atrybuty:	
Metody:	
Realizowane wymagania:	WF007, WI004
Priorytet:	ważne

CN006	ClassNode
Opis:	Klasa reprezentuje wierzchołek OWL Class.
Klasy nadrzędne:	Node
Atrybuty:	
Metody:	
Realizowane wymagania:	WF004, WF005, WI004
Priorytet:	ważne

CN007	ComplementOfNode
-------	------------------

Opis:	Klasa reprezentuje wierzchołek klas anonimowych OWL będących wynikiem dopełnienia (OWL ComplementOf).
Klasy nadrzędne:	Node
Atrybuty:	
Metody:	
Realizowane wymagania:	WF006, WF007, WI004
Priorytet:	ważne

CN008	DataTypeNode
Opis:	Klasa reprezentuje wierzchołek OWL DataType.
Klasy nadrzędne:	Node
Atrybuty:	
Metody:	
Realizowane wymagania:	WF004, WI04
Priorytet:	ważne

CN009	DifferentNode
Opis:	Klasa reprezentuje wierzchołek oznaczający relację DifferentFrom lub AllDifferent pomiędzy wystąpieniami klas (OWL Individual).
Klasy nadrzędne:	Node
Atrybuty:	
Metody:	
Realizowane wymagania:	WF006, WF007, WI004
Priorytet:	ważne

CN010	FunctionalPropertyNode
Opis:	Klasa reprezentuje wierzchołek oznaczający, że dane OWL Property to FunctionalProperty.
Klasy nadrzędne:	InformationNode
Atrybuty:	
Metody:	
Realizowane wymagania:	WF006, WF007, WI004
Priorytet:	ważne

CN011	IndividualNode
Opis:	Klasa reprezentuje wierzchołek instancji OWL Individual.
Klasy nadrzędne:	Node
Atrybuty:	
Metody:	
Realizowane wymagania:	WF004, WI004
Priorytet:	ważne

CN012	InformationNode
Opis:	Klasa ta jest klasą nadrzędną, dla klas wierzchołków reprezentujących informacje o różnych właściwościach OWL Property.
Klasy nadrzędne:	Node
Atrybuty:	

Metody:	
Realizowane wymagania:	WF010, WI004
Priorytet:	ważne

CN013	IntersectionOfNode
Opis:	Klasa reprezentuje wierzchołek klas anonimowych OWL będących wynikiem przecięcia (OWL IntersectionOf).
Klasy nadrzędne:	AnonymousNode
Atrybuty:	
Metody:	
Realizowane wymagania:	WF005, WI004
Priorytet:	ważne

CN014	inverseFunciotnalPropertyNode
Opis:	Klasa reprezentuje wierzchołek oznaczający, że dane OWL Property to InverseFunctionalProperty.
Klasy nadrzędne:	InformationNode
Atrybuty:	
Metody:	
Realizowane wymagania:	WF007, WI004
Priorytet:	ważne

CN015	MaxCardinalityValueNode
Opis:	Klasa reprezentuje wierzchołek ograniczenia kardynalności OWL MaxCardinality.
Klasy nadrzędne:	CardinalityValueNode
Atrybuty:	
Metody:	
Realizowane wymagania:	WF007, WI004
Priorytet:	ważne

CN016	MinCardinalityValueNode
Opis:	Klasa reprezentuje wierzchołek ograniczenia kardynalności OWL MinCardinality.
Klasy nadrzędne:	CardinalityValueNode
Atrybuty:	
Metody:	
Realizowane wymagania:	WF007, WI004
Priorytet:	ważne

CN017	NothingNode
Opis:	Klasa reprezentuje wierzchołek OWL Nothing.
Klasy nadrzędne:	Node
Atrybuty:	
Metody:	
Realizowane wymagania:	WF004, WF005, WI004
Priorytet:	ważne

CN018	OneOfNode
Opis:	Klasa reprezentuje wierzchołek klas anonimowych OWL reprezentujących 1 z klas określonego zbioru (wynik OWL OneOf).
Klasy nadrzędne:	AnonymousClassNode
Atrybuty:	
Metody:	
Realizowane wymagania:	WF005, WF006, WI004
Priorytet:	ważne

CN019	PropertyNode
Opis:	Klasa reprezentuje wierzchołek OWL Property.
Klasy nadrzędne:	Node
Atrybuty:	
Metody:	
Realizowane wymagania:	WF004, WF007, WI004
Priorytet:	ważne

CN020	SameAsNode
Opis:	Klasa reprezentuje wierzchołek oznaczający relację OWL SameAs pomiędzy wystąpieniami klas (OWL Individual).
Klasy nadrzędne:	InformationNode
Atrybuty:	
Metody:	
Realizowane wymagania:	WF005, WF006, WI004
Priorytet:	ważne

CN021	SomeValuesFromPropertyNode
Opis:	Klasa reprezentuje wierzchołek, będący OWL Property typu SomeValuesFrom.
Klasy nadrzędne:	PropertyNode
Atrybuty:	
Metody:	
Realizowane wymagania:	WF005, WF006, WI004
Priorytet:	ważne

CN022	SymmetricPropertNode
Opis:	Klasa reprezentuje wierzchołek oznaczający, że dane OWL Property to SymmetricProperty.
Klasy nadrzędne:	InformationNode
Atrybuty:	
Metody:	
Realizowane wymagania:	WF007, WI004
Priorytet:	ważne

CN023	ThingNode
Opis:	Klasa reprezentuje wierzchołek OWL Thing.
Klasy nadrzędne:	Node
Atrybuty:	

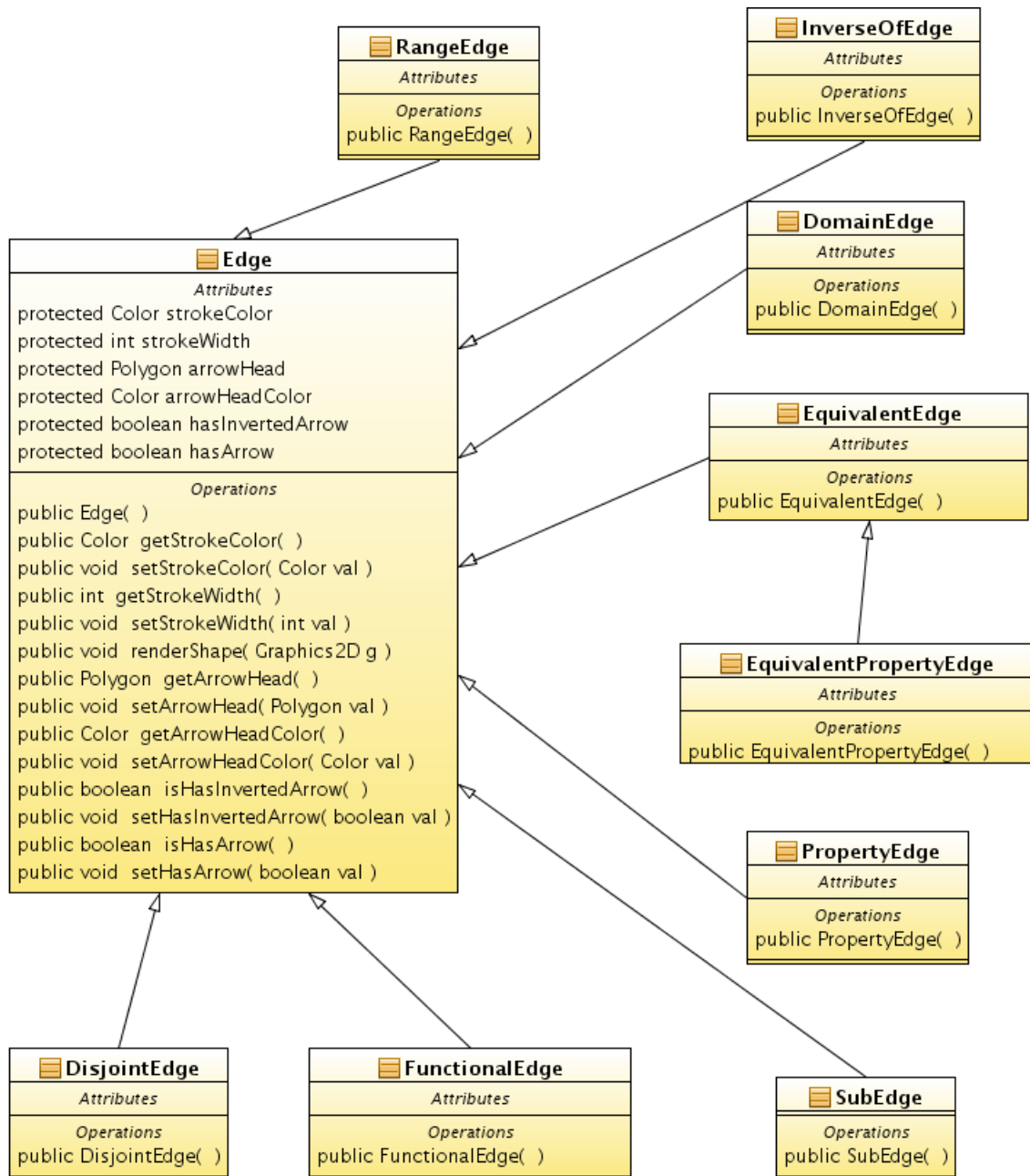
Metody:	
Realizowane wymagania:	WF004, WF005, WI004
Priorytet:	ważne

CN024	TransitivePropertyNode
Opis:	Klasa reprezentuje wierzchołek oznaczający, że dane OWL Property to TransitiveProperty.
Klasy nadrzędne:	InformationNode
Atrybuty:	
Metody:	
Realizowane wymagania:	WF006, WF007, WI004
Priorytet:	ważne

CN025	UnionOfNode
Opis:	Klasa reprezentuje wierzchołek klas anonimowych OWL będących wynikiem unii (OWL UnionOf).
Klasy nadrzędne:	AnonymousNode
Atrybuty:	
Metody:	
Realizowane wymagania:	WF005, WF006, WI004
Priorytet:	ważne

4 Pakiet edges

4.1 Diagram



4.2 Opis klasy

CE001	Edge
Opis:	Klasa reprezentująca prostą krawędź na grafie. Jest nadklasą dla pozostałych klas krawędzi.
Klasy nadrzędne:	

Atrybuty:	<ul style="list-style-type: none"> • Color strokeColor • int strokeWidth • boolean hasArrow • boolean hasInvertedArrow • Polygon arrowHead • Color arrowHeadColor
Metody:	<ul style="list-style-type: none"> • getStrokeColor () • setStrokeColor (Color val) • getStrokeWidth () • setStrokeWidth (int val) • getArrowHead() • setArrowHead(Polygon arrowHead) • isHasArrow() • setHasArrow(boolean hasArrow) • isHasInvertedArrow() • setHasInvertedArrow(boolean hasInvertedArrow) • getArrowHeadColor() • setArrowHeadColor(Color arrowHeadColor)
Realizowane wymagania:	WF006, WF007, WI004
Priorytet:	bardzo ważne

CE002	DisjointEdge
Opis:	Klasa reprezentująca krawędź oznaczającą rozłączność klas (OWL Disjoint).
Klasy nadrzędne:	Edge
Atrybuty:	
Metody:	
Realizowane wymagania:	WF006, WF007, WI004
Priorytet:	ważne

CE003	DomainEdge
Opis:	Klasa reprezentująca krawędź łączącą Property z klasą właściwości OWL DomainOf.
Klasy nadrzędne:	Edge
Atrybuty:	
Metody:	
Realizowane wymagania:	WF006, WF007, WI004

Priorytet:	ważne
------------	-------

CE004	EquivalentEdge
Opis:	Klasa reprezentująca krawędź oznaczającą równoznaczność (OWL Equivalent).
Klasy nadrzędne:	Edge
Atrybuty:	
Metody:	
Realizowane wymagania:	WF006, WF007, WI004
Priorytet:	ważne

CE005	EquivalentPropertyEdge
Opis:	Klasa reprezentująca krawędź oznaczającą równoznaczność OWL Property (OWL EquivalentProperty).
Klasy nadrzędne:	EquivalentEdge
Atrybuty:	
Metody:	
Realizowane wymagania:	WF006, WF007, WI004
Priorytet:	ważne

CE006	FunctionalEdge
Opis:	Klasa reprezentująca krawędź łączącą wierzchołki InformationNode(CN012) z OWL Property, którego dotyczy.
Klasy nadrzędne:	Edge
Atrybuty:	
Metody:	
Realizowane wymagania:	WF006, WF007, WI004
Priorytet:	ważne

CE007	InverseOfEdge
Opis:	Klasa reprezentująca krawędź oznaczającą odwrotność (OWL InverseOf).
Klasy nadrzędne:	Edge
Atrybuty:	
Metody:	
Realizowane wymagania:	WF006, WF007, WI004
Priorytet:	ważne

CE008	PropertyEdge
Opis:	Klasa reprezentująca krawędź oznaczającą relację między Property a klasą.
Klasy nadrzędne:	Edge
Atrybuty:	
Metody:	
Realizowane wymagania:	WF006, WF007, WI004
Priorytet:	ważne

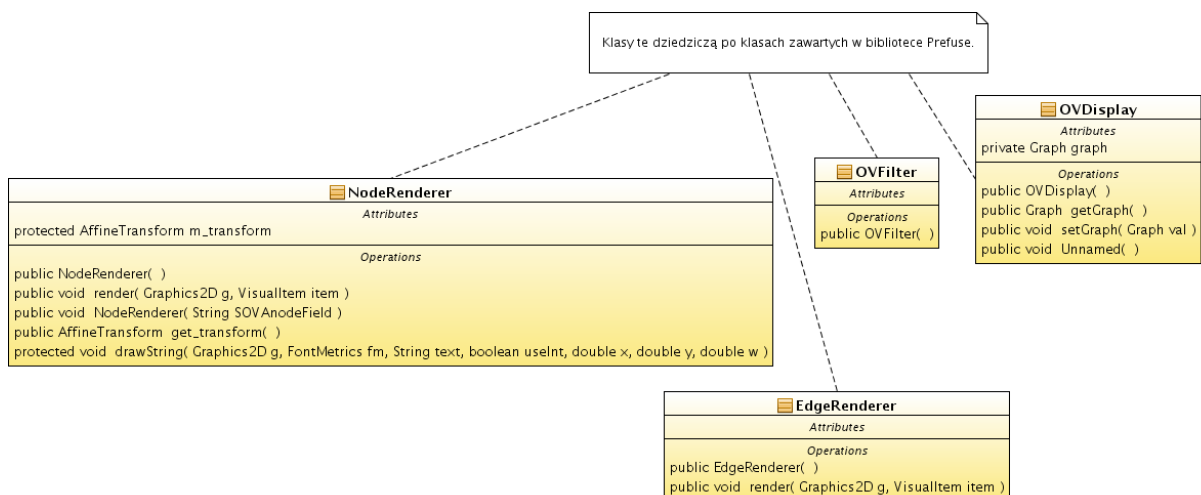
CE009	RangeEdge
-------	-----------

Opis:	Klasa reprezentująca na grafie krawędź łączącą Property z klasą właściwości OWL Range.
Klasy nadrzędne:	Edge
Atrybuty:	
Metody:	
Realizowane wymagania:	WF006, WF007, WI004
Priorytet:	ważne

CE010	SubEdge
Opis:	Klasa reprezentująca krawędź związku OWL SubClass pomiędzy klasami.
Klasy nadrzędne:	Edge
Atrybuty:	
Metody:	
Realizowane wymagania:	WF006, WF007, WI004
Priorytet:	ważne

5 Pakiet visualization

5.1 Diagram



5.2 Opis klasy

CV001	EdgeRenderer
Opis:	Klasa przeciążająca metody renderowania krawędzi grafu z biblioteki prefuse.
Klasy nadrzędne:	prefuse.render.EdgeRenderer
Atrybuty:	
Metody:	<ul style="list-style-type: none"> render(Graphics2D g, VisualItem item) - metoda renderująca krawędź
Realizowane wymagania:	WF001, WF008, WI004
Priorytet:	ważne

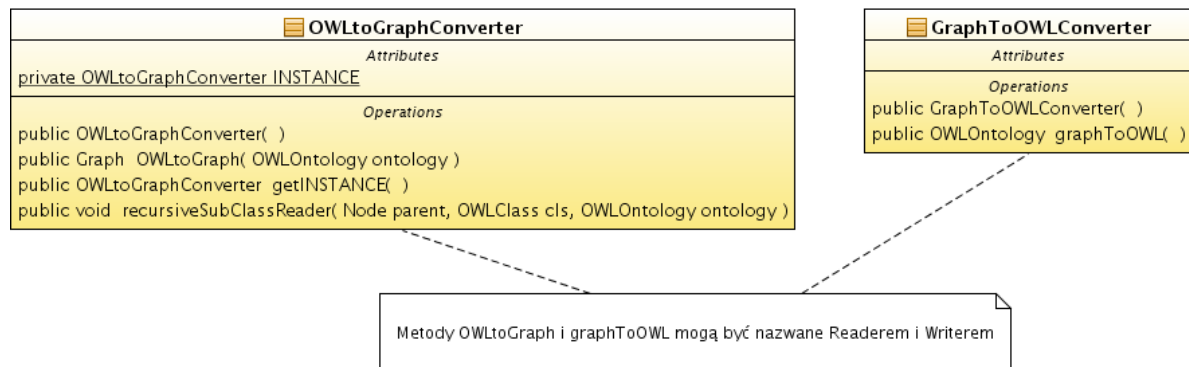
CV002	NodeRenderer
Opis:	Klasa przeciążająca metody renderowania wierzchołków grafu z biblioteki prefuse.
Klasy nadrzędne:	prefuse.render.LabelRenderer
Atrybuty:	
Metody:	<ul style="list-style-type: none"> • render (Graphics2D g, VisualItem item) - metoda renderująca wierzchołek • drawString(Graphics2D g, FontMetrics fm, String text, boolean useInt, double x, double y, double w) - metoda wypisująca na wierzchołku String
Realizowane wymagania:	WF001, WF008, WI004
Priorytet:	ważne

CV003	OVDDisplay
Opis:	Klasa tworząca obiekt JComponent do umieszczenia na okienku JAVA zawierający wygenerowany graf z wizualizacją
Klasy nadrzędne:	prefuse.Display
Atrybuty:	<ul style="list-style-type: none"> • Graph graph - obiekt typu prefuse.data.graph zawierający dane o grafie do wyświetlenia.
Metody:	<ul style="list-style-type: none"> • getGraph() - zwraca graf z wyświetlanymi danymi • setGraph(Graph graph) - nadpisuje obecny graf podanym • generateGraphFromOWL(OWLontology ont) - wpisuje do klasy obiekt Grpah wygenerowany na podstawie ontologii
Realizowane wymagania:	WF001, WF002, WF008, WI004
Priorytet:	ważne

CV004	OVFilter
Opis:	Klasa zawierająca filtry służące do wyświetlania danych w różnych zakresach
Klasy nadrzędne:	
Atrybuty:	
Metody:	
Realizowane wymagania:	WF001, WF008, WI004
Priorytet:	ważne

6 Pakiet graph

6.1 Diagram



6.2 Opis klasy

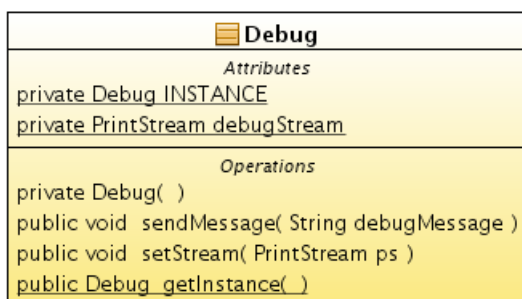
CG001	GraphToOWLConverter
Opis:	Klasa zawierająca metody pozwalające na przetwarzanie obiektów grafów z prefuse na obiekty OWL API. Klasa jest singletonem.
Klasy nadrzędne:	
Atrybuty:	<ul style="list-style-type: none"> INSTANCE - instancja klasy GraphToOWLConverter
Metody:	<ul style="list-style-type: none"> getInstance() - zwraca instancję klasy GraphToOWL(OWLOntology ontology) -Zamienia graf z biblioteki prefuse na ontologię zapisana w OWL API.
Realizowane wymagania:	WD001, WI004
Priorytet:	ważne

CG002	OWLtoGraphConverter
Opis:	Klasa zawierająca metody pozwalające na przetwarzanie obiektów OWL API na obiekty prefuse. Klasa jest singletonem.
Klasy nadrzędne:	
Atrybuty:	<ul style="list-style-type: none"> INSTANCE - instancja klasy GraphToOWLConverter

Metody:	<ul style="list-style-type: none"> • getInstance() - zwraca instancję klasy • recursiveSubClassReader(Node parent, OWLClass cls, OWLOntology ontology) - wczytuje do grafu OWL wszystkie klasy wraz z ich podklasami. • OWLToGraph(OWLOntology ontology) -Zamienia ontologię w OWL API na graf z biblioteki prefuse.
Realizowane wymagania:	WD001, WI004
Priorytet:	ważne

7 Pakiet utils

7.1 Diagram



7.2 Opis klasy

CU001	Debug
Opis:	Klasa do użycia przy debugowaniu, zapewnia strumień z błędami zwracanymi przez bibliotekę. Klasa jest singletonem.
Klasy nadrzędne:	
Atrybuty:	<ul style="list-style-type: none"> • INSTANCE - instancja klasy Debug • Debug - Strumień do którego wpisywane są informacje potrzebne do debugowania
Metody:	<ul style="list-style-type: none"> • getInstance() - zwraca instancję klasy • setStream(PrintStream ps) - ustawia podany strumień jako strumień na który zwracane będą błędy • sendMessage(String s) - wysła wiadomość na strumień do debugowania, jeżeli został wcześniej podpięty za pomocą funkcji setStream
Realizowane wymagania:	WF006, WF007, WI004
Priorytet:	bardzo ważne